Program #1
Due: Monday Jan 26th, 2015 at 11:30PM

Instructor	Dr. Stephen Perkins
Office Location	ECSS 4.403
Email Address	stephen.perkins@utdallas.edu
Office Hours	Monday and Wednesday before class (7:30pm - 8:25pm) And by appointment
TA	TBD

Purpose

Demonstrate the ability to use text editors in the shell. Demonstrate the ability to create and execute shell scripts. Demonstrate the ability to create and compile multi-file C++ programs. Demonstrate the ability to access the command line arguments from within a C++ program. Demonstrate proficiency in logging your console output using the "screen –L" command.

Assignment

Document all the shell work done to complete the assignment using the "screen –L" program or putty logging. This will keep a log of all your shell work. You will need to turn in the output files from "*screen –L*"/putty. See "**Deliverables**" below.

With console logging enabled, write a program that consists of two C++ source files. The first file contains the main() routine and the second file contains a C++ procedure.

> From within **main**(), you must display the count of command line arguments to *stdout*. Further, you must display the value of each command line argument to *stdout*. Finally, you must call the procedure that is defined in the second file.
>
> From within the procedure defined in the second file, you must display a log message that states you are inside the procedure. This log message must be sent to *stderr*.

You are then to create a shell script for compiling your code. The script will individually compile each file and generate the intermediate object file. Once the object files are created, the script will call the linker and link the object files into an executable. Your shell script should echo what it is doing at each step to *stdout*.

You are then to create a second shell script that will run your program several times with varying command line arguments. Each time you call your program, you should **append** the *stdout* to an output file called ***stdout.log*** and you should **append** the *stderr* to an output file called ***stderr.log***. Your shell script should echo what it is doing at each step to *stdout.*

Deliverables

You must submit your homework through ELearning. You must include your .cc files, shell scripts, stdout.log, stderr.log, and your shell logging output file/files generated by the "screen –L" command or putty logging. All source files and shell scripts need to have your name, email, and course number commented at the top.

Notes

If the compiler complains that /tmp is full ("No space left on device"), then set your TMPDIR environment variable to /scratch. You can do this in your bash shell and in your shell scripts like this

```
TMPDIR=/scratch; export TMPDIR
```

Use the "screen –L" command to capture all your shell work. Output is logged in a file named screenlog.0. You may exit the "screen" program by typing CTRL-D. Be careful to rename/preserve the screenlog.0 file if you restart "screen". It will overwrite the file.

Alternately, use putty logging to log to a local file.

All editing must be done within the shell using a text editor.

No late homework is accepted.

Example Output

```
$> compile.sh

Setting TEMPDIR environment variable to /scratch
Compiling file1.cc
Compiling file12.cc
Linking files to create executable hw1
Done



$> run.sh

Running 'hw1' with 0 arguments:
     stdout appended to stdout.txt
     stderr appended to stderr.txt

Running 'hw1' with 1 argument:
     stdout appended to stdout.txt
     stderr appended to stderr.txt

Running 'hw1' with 5 arguments:
     stdout appended to stdout.txt
     stderr appended to stderr.txt
```

```
$> more stdout.txt

argc was: 1
./hw1
Done!
argc was: 2
./hw1
abc
Done!
argc was: 6
./hw1
a
b
c
d
e
Done!
```

```
$> more stderr.txt

Inside proc1() as stderr
Inside proc1() as stderr
Inside proc1() as stderr
```